REMARKS

Claims 1-36 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Provisional Double Patenting Rejection:

The Examiner provisionally rejected claims 1-36 under the judiciary created doctrine of double patenting over claims 1-56 of copending Application No. 10/618,810. If this rejection becomes non-provisional, Applicants will either submit a terminal disclaimer or present arguments traversing the rejection.

Section 103(a) Rejections:

The Examiner rejected claims 1-8 and 10-19, 25-30 and 36 under 35 USC § 103(a) as being unpatentable over Hagersten et al. (U.S. Patent 5,983,326) (hereinafter "Hagersten"), claim 9 as being unpatentable over Hagersten in view of Fowler (U.S. Patent 4,502,116), claims 20-24 and 31-35 as being unpatentable over Hagersten in view of Fowler and further in view of Klein et al. (U.S. Patent 6,728,958) (hereinafter "Klein"). Applicants respectfully traverse these rejections for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement a transaction manager*. The Examiner refers to the fact that Hagersten describes memory that includes code for use by its processors. However, the code in Hagersten's memory is not described as executable by the one or more processors to implement a transaction manager. In fact, the functionality in Hagersten relied upon by the Examiner to teach the transaction manager of claim 1 is part of system interface 24, which is separate from memory 22 and processors 16. *See* Hagersten, col. 7, lines 44-53 cited by the Examiner.

Hagersten's home agent and system interface is not described as being implemented by program instructions stored in memory 22 or any other memory.

Further regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *a transaction manager configured to manage a plurality of transactions initiated by one or more applications*. The Examiner refers to col. 1, lines 19-22 from the related art section of Hagersten. This portion of Hagersten refers to components of a computing task being distributed among multiple processors. This has absolutely nothing to do with transactions initiated by an application. An application is a software-level entity. In contrast, the transactions in Hagersten are hardware-level memory transactions initiated by processors and communicated on a hardware bus. Hagersten's transactions are a completely different type of transaction from transactions that are initiated by software applications. Therefore, the Examiner's assertion is not supported by the cited art. **Moreover, Applicants note that in the previous Action dated November 30, 2007, the Examiner admitted that Hagersten does not teach this limitation of claim 1.**

Further regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *wherein each transaction comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction*. The Examiner refers to col. 8, lines 20-34, of Hagersten. However, Hagersten does not teach that each individual transaction comprises a **plurality** of operations. To the contrary, the portion of Hagersten cited by the Examiner states that a transaction includes at most a single memory operation. Also, the portion of Hagersten cited by the Examiner says nothing of **each individual** transaction comprising a **plurality** of operations that are required to be committed to the one or more data sources atomically for each respective transaction.

Further in regard to claim 1, Hagersten in view of Fowler fails to disclose *wherein while paused, the transaction manager* ***does not allow any of the plurality of transactions*** *managed by the transaction manager to complete.* Hagersten teaches the

blocking of one transaction request while another transaction request is being serviced. Hagersten states in column 7, lines 46-49, "However, a blocking mechanism is employed to prevent the servicing of a particular coherent transaction request if another transaction request corresponding to the same coherency unit is currently being service by the system interface." Hagersten, whether or not combined with Fowler, does not teach pausing **all** transactions managed by the transaction manager in response to a pause request as required by Applicants' claim 1. **To the contrary, Hagersten explicitly allows the current transaction to complete while blocking one or more other transactions.**

Further regarding claim 1, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *pause the plurality of transactions managed by the transaction manager* ***in response to a pause request*** *to pause the transaction manager, wherein while paused, the transaction manager does not allow any of the plurality of transactions managed by the transaction manager to complete.* The Examiner cites Hagersten, column 5, lines 55-64; column 7, lines 44-53, and column 17, lines 7-9. These passages teach one or more queues configured to receive transaction requests from processing nodes. The passages also teach a home agent control unit configured to receive and service transaction requests and a transaction blocking unit coupled to the queues and the home agent. For example, column 5, lines 60 through column 6, line 4 states:

> The transaction blocking unit is configured to block selected transactions if another transaction request to a common coherency unit is currently being serviced by the home agent control unit. The transaction blocking unit is further configured to allow servicing of a given transaction request to a particular coherency unit if a second transaction request to the particular coherency unit is currently being serviced by the home agent control unit and if the second transaction request does not cause ownership of the particular coherency unit and if the second transaction request and the given transaction request are the same transaction type.

Applicants assert that the transaction blocking unit of Hagersten clearly does not block transactions **in response to a pause request** as required by Applicants' claim 1. Instead, the transaction blocking unit, blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. The

transaction blocking unit is not **responding to a request** to pause, but rather is responding to the detection of a second transaction request arriving when the home agent is servicing a first transaction request.

Further in regard to claim 1, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose that *the transaction manager is configured to resume the plurality of transactions managed by the transaction manager in response to a resume request*. The Examiner cites column 17, lines 9-12, which states, "…subsequent requests involving the coherency unit are not performed until the coherency activity corresponding to the coherency request is completed." Applicants assert that Hagersten does not teach resuming the transaction manager in response to a resume request. Instead, Hagersten teaches activity is resumed when the coherency unit completes the current task. Clearly there is no resume request taught by Hagersten. Further, the Examiner admits Hagersten fails to teach generating a pause request and a resume request in paragraph 24 of the Instant Office Action. The Examiner states, "Hagersten does not explicitly teach generating a pausing request and a resuming request." In addition, Fowler fails to teach a pause request or a resume request.

The Examiner admits Hagersten fails to teach receiving a pause request and a resume request and relies on Fowler. Fowler is directed towards a system and method to implement a synchronization interface circuit whereby the pausing of one processor in a system with multiple processors causes other interconnected processors to pause. The Examiner cites Fowler, column 5, lines 40-53 and column 8, lines 52-56. Column 5, lines 40-43 state, "The control section 80 of the Subsystem Synchronization Interface circuit provides users with **toggle switches** to configure the pause/resume synchronization of the multiprocessor system." Applicants assert that a manual toggle switch that causes processors to pause has absolutely nothing to do with the teachings of Hagersten or with the limitations of claim 1. Neither Hagersten or Fowler teach pausing transactions managed by the transaction manager in response to a pause request to pause the transaction manager. Fowler does not teach a transaction manager and does not teach

receiving a pause request to pause a transaction manager. Nor does Fowler teach resuming transactions managed by a transaction manager in response to a resume request.

Furthermore, the Examiner's proposed combination makes no sense and would be inoperable. The transactions in Hagersten are paused and resumed in response to detection of very specific events. For example, Hagersten's transaction blocking unit blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. It would make absolutely no sense whatsoever to modify Hagersten to instead pause a transaction in response to **a signal from a manual toggle switch** as taught by Fowler. In fact, given that Hagersten's transactions are high speed hardware bus communications, it would be impossible to pause such a transaction with a manual toggle switch, as taught by Fowler. **Furthermore, the pause request in Fowler is to pause the entire processor, not a single transaction.** The Examiner's proposed modification of Hagersten would destroy the intended operation of Hagersten's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

Regarding claim 14, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *memory coupled to the one or more processors and configured to store program instructions executable by the one or more processors to implement one or more application servers, wherein each one or more application servers is configured to run one or more applications ...and provide one or more transaction managers*. The Examiner refers to the fact that Hagersten describes memory that includes code for use by its processors. However, the code in Hagersten's memory is not described as executable by the one or more processors to implement an application server(s) that runs an application(s) and provides a transaction manager(s). In fact, Hagersten has absolutely nothing to do with application servers. Furthermore, the functionality in Hagersten relied upon by the Examiner to teach the one or more transaction managers of claim 14 is part of system interface 24 which is separate from memory 22 and processors 16. *See*

Hagersten, col. 7, lines 44-53 cited by the Examiner. Hagersten's home agent and system interface is not described as being implemented by program instructions stored in memory 22.

Further regarding claim 14, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *wherein each of the one or more transactions comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for the transaction*. The Examiner refers to col. 8, lines 20-34, of Hagersten. However, Hagersten does not teach that each transaction comprises a plurality of operations. To the contrary, the portion of Hagersten cited by the Examiner states that a transaction includes at most a single memory operation. Also, the portion of Hagersten cited by the Examiner says nothing of each transaction comprising a plurality of operations that are required to be committed to the one or more data sources atomically for each respective transaction.

Further regarding claim 14, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose that *one of the transaction managers is configured to pause a corresponding one or more transactions in response to a pause request*. The Examiner cites Hagersten, column 5, lines 55-64; column 7, lines 44-53, and column 17, lines 7-9. These passages teach one or more queues configured to receive transaction requests from processing nodes. The passages also teach a home agent control unit configured to receive and service transaction requests and a transaction blocking unit coupled to the queues and the home agent. For example, column 5, lines 60 through column 6, line 4 states:

> The transaction blocking unit is configured to block selected transactions if another transaction request to a common coherency unit is currently being serviced by the home agent control unit. The transaction blocking unit is further configured to allow servicing of a given transaction request to a particular coherency unit if a second transaction request to the particular coherency unit is currently being serviced by the home agent control unit and if the second transaction request does not cause ownership of the particular coherency unit and if the second transaction request and the given transaction request are the same transaction type.

Applicants assert that the transaction blocking unit of Hagersten clearly does not block transactions **in response to a pause request** as required by Applicants' claim 14. Instead, the transaction blocking unit, blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. The transaction blocking unit is not **responding to a request** to pause, but rather is responding to the detection of a second transaction request arriving when the home agent is servicing a first transaction request. Nowhere does Hagersten or Fowler, taken in combination or singularly, teach *a transaction manager configured to pause a corresponding one or more transactions* ***in response to a pause request.***

Further in regard to claim 14, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose that *the transaction manager is configured to resume the corresponding one or more transactions in response to a resume request*. The Examiner cites column 17, lines 9-12, which states, "…subsequent requests involving the coherency unit are not performed until the coherency activity corresponding to the coherency request is completed." Applicants assert that Hagersten does not teach resuming a transaction in response to a resume request. Instead, Hagersten teaches activity is resumed when the coherency unit completes the current task. Clearly there is no resume request taught by Hagersten. Further, the Examiner admits Hagersten fails to teach generating a pause request and a resume request in paragraph 9 of the Instant Office Action. The Examiner states, "Hagersten does not explicitly teach generating a pausing request and a resuming request." In addition, Fowler fails to teach a pause request or a resume request.

Furthermore, the Examiner's proposed combination makes no sense and would be inoperable. The transactions in Hagersten are paused and resumed in response to detection of very specific events. For example, Hagersten's transaction blocking unit blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. It would make absolutely no sense whatsoever to modify Hagersten to instead pause a transaction in response to **a signal from a manual toggle switch** as taught by Fowler. In fact, given that Hagersten's

transactions are high speed hardware bus communications, it would be impossible to pause such a transaction with a manual toggle switch, as taught by Fowler. **Furthermore, the pause request in Fowler is to pause the entire processor, not a single transaction.** The Examiner's proposed modification of Hagersten would destroy the intended operation of Hagersten's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

Regarding claim 15, contrary to the Examiner's assertion, Hagersten in view of Fowler fails to disclose, *wherein each transaction managed by the transaction manager comprises a plurality of operations to one or more data sources that are required to be committed to the one or more data sources atomically for each respective transaction*. The Examiner refers to col. 8, lines 20-34, of Hagersten. However, Hagersten does not teach that each transaction comprises a plurality of operations. To the contrary, the portion of Hagersten cited by the Examiner states that a transaction includes at most a single memory operation. Also, the portion of Hagersten cited by the Examiner says nothing of each transaction comprising a plurality of operations that are required to be committed to the one or more data sources atomically for each respective transaction.

Further regarding claim 15, contrary to the Examiner's assertion, Hagersten fails to disclose, *generating a request to pause a transaction manager; pausing the transaction manager in response to said request*. The Examiner relies on Hagersten, column 5, lines 55-64; column 7, lines 44-53; column 17, lines 7-9. These passages teach one or more queues configured to receive transaction requests from processing nodes. The passages also teach a home agent control unit configured to receive and service transaction requests and a transaction blocking unit coupled to the queues and the home agent. (See the quoted passages above in the remarks for claim 1.) Hagersten and Fowler do not teach generating a pause request to pause a transaction manager, or pausing and resuming the transaction manager *in response to a request*. As discussed

above in regard to claims 1 and 14, the home agent in Hagersten blocks transactions based on detecting the state of an existing transaction, **not in response to any request**.

Furthermore, the Examiner's proposed combination makes no sense and would be inoperable. The transactions in Hagersten are paused and resumed in response to detection of very specific events. For example, Hagersten's transaction blocking unit blocks a second transaction if a first transaction request (to a common coherency unit) is currently being serviced by the home agent. It would make absolutely no sense whatsoever to modify Hagersten to instead pause a transaction in response to **a signal from a manual toggle switch** as taught by Fowler. In fact, given that Hagersten's transactions are high speed hardware bus communications, it would be impossible to pause such a transaction with a manual toggle switch, as taught by Fowler. **Furthermore, the pause request in Fowler is to pause the entire processor, not a single transaction.** The Examiner's proposed modification of Hagersten would destroy the intended operation of Hagersten's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

Similar arguments as presented above for claim 15 apply to claim 26 as well.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-15100/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: May 5, 2008